

-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  **Flex**
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  PL/SQL
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  T-SQL
-  VB.NET
-  VB6
-  XML



# Flex static code analysis

Unique rules to find Bugs, Security Hotspots, and Code Smells in your FLEX code

- All rules76
-  Vulnerability5
-  Bug9
-  Security Hotspot1
-  Code Smell61

Tags

Search by name...

Security.allowDomain(...) should only be used in a tightly focused manner		Vulnerability
The flash.system.Security.exactSettings property should never be set to false		Vulnerability
Dynamic classes should not be used		Code Smell
"LocalConnection" should be configured to narrowly specify the domains with which local connections to other Flex application are allowed		Vulnerability
"default" clauses should be first or last		Code Smell
Event types should be defined in metadata tags		Code Smell
Event names should not be hardcoded in event listeners		Code Smell
The special "star" type should not be used		Code Smell
Variables of the "Object" type should not be used		Code Smell
Methods should not be empty		Code Smell
Constant names should comply with a naming convention		Code Smell
All branches in a conditional structure should not have exactly the same implementation		Bug
Classes that extend "Event" should		

## Two branches in a conditional structure should not have exactly the same implementation

Analyze your code

-  Code Smell
-  Major?
-  design suspicious

Having two cases in a switch statement or two branches in an if chain with the same implementation is at best duplicate code, and at worst a coding error. If the same logic is truly needed for both instances, then in an if chain they should be combined, or for a switch, one should fall through to the other.

### Noncompliant Code Example











```
switch (i) {
  case 1:
    doFirstThing();
    doSomething();
    break;
  case 2:
    doSomethingDifferent();
    break;
  case 3:  // Noncompliant; duplicates case 1's implemen
    doFirstThing();
    doSomething();
    break;
  default:
    doTheRest();
}

if (a >= 0 && a < 10) {
  doFirstThing();
  doTheThing();
}
else if (a >= 10 && a < 20) {
  doTheOtherThing();
}
else if (a >= 20 && a < 50) {
  doFirstThing();
  doTheThing();  // Noncompliant; duplicates first condi
}
else {
  doTheRest();
}
```

### Exceptions

Blocks in an if chain that contain a single line of code are ignored, as are blocks in a switch statement that contain a single line of code with or without a following break.

```
if (a == 1) {
  doSomething();  //no issue, usually this is done on pu
} else if (a == 2) {
  doSomethingElse();
} else {
  doSomething();
}
```

<div>override "Event.clone()"</div> <div> Bug</div>
<div>Constructors should not dispatch events</div> <div> Bug</div>
<div>"ManagedEvents" tags should have companion "Event" tags</div> <div> Bug</div>
<div>Objects should not be instantiated inside a loop</div> <div> Code Smell</div>
<div>Two branches in a conditional structure should not have exactly the same implementation</div> <div> Code Smell</div>
<div>Constructor bodies should be as lightweight as possible</div> <div> Code Smell</div>
<div>Only "while", "do" and "for" statements should be labelled</div> <div> Code Smell</div>
<div>Statements, operators and keywords specific to ActionScript 2 should not be used</div> <div> Code Smell</div>
<div>"for" loop stop conditions should be invariant</div> <div> Code Smell</div>
<div>Unused function parameters should be removed</div> <div> Code Smell</div>

But this exception does not apply to `if` chains without `else`-s, or to `switch`-es without default clauses when all branches have the same single line of code. In case of `if` chains with `else`-s, or of `switch`-es with default clauses, rule `{rule:flex:S3923}` raises a bug.

```
if (a == 1) {
    doSomething(); //Noncompliant, this might have been d
} else if (a == 2) {
    doSomething();
}
```

Available In: